

Projet VALMEM

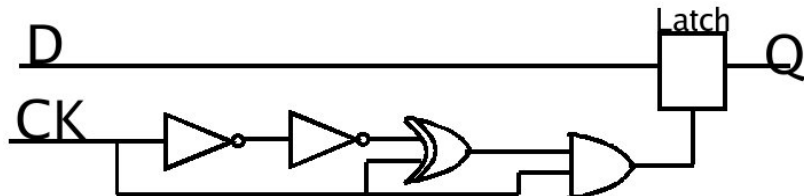
# Génération de contraintes sur le nouveau circuit

Étienne André

Laboratoire Spécification et Vérification

# Rappel du système

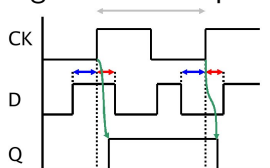
- Cinq éléments :
  - ▶ Deux portes « non »
  - ▶ Une porte « ou exclusif »
  - ▶ Une porte « et »
  - ▶ Une bascule à niveau (« latch »)



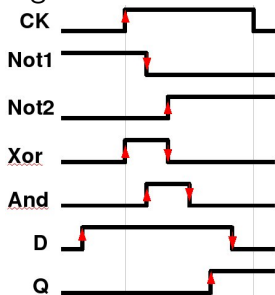
- On note **E** la sortie de la porte « non » (entrée activant le latch)

# Signaux

- Signaux fournis par Rémy

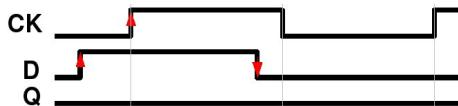


- Signaux aux différentes portes



# Modélisation

- Modélisation des portes
  - ▶ Seules les transitions marquées en rouge sont modélisées
  - ▶ Ex : l'automate de la porte « not 1 » ne fait que produire un front descendant
- Définition d'un mauvais état
  - ▶ Un cycle d'horloge complet a eu lieu
  - ▶ Q n'a pas effectué de front montant



- Utilisation d'intervalles
  - ▶ Expression de toutes les durées du système sous forme d'intervalles
  - ▶ Exemple : durée de passage du latch à 1  
 $dLatch\_l < dLatch\_u$

# Méthode d'obtention des contraintes

- Utilisation de HyTech
  - ▶ Départ de l'état final mauvais (pas de front montant sur Q)
  - ▶ Calcul des états accessibles en arrière
  - ▶ Intersection non vide avec l'état initial : contre-exemple
  - ▶ Obtention d'un ensemble de contraintes
  - ▶ Ajout de la négation de l'une des contraintes au système et retour à la première étape
- Lorsque l'intersection entre l'état initial et les états accessibles en arrière depuis le mauvais état est vide :
  - ▶ Obtention d'un ensemble de contraintes assurant le bon fonctionnement du système
- Remarque : le contre-exemple retenu est minimal
  - ▶  $\text{Pre}^n(\text{Final}) \cap \text{Init} \neq \emptyset$
  - ▶  $\text{Pre}^{n-1}(\text{Final}) \cap \text{Init} = \emptyset$

# Contraintes obtenues

- Suppression des contraintes redondantes
  - ▶ Tests successifs de suppression de contraintes en vérifiant que l'intersection entre l'état initial et les états accessibles depuis le mauvais état reste vide

- Ensemble final de contraintes non redondantes :

$$dHold > dAndUp2\_u + dLatchUp\_u$$

$$dAndUp2\_u + dLatchUp\_u < dNot1Down\_l + dNot2Up\_l + dXorDown1Up\_l + dAndDown1\_l$$

- Remarque

- ▶ D'autres jeux de contraintes auraient pu être obtenus
  - ★ Choix d'un autre contre-exemple
  - ★ Choix d'une autre contrainte à nier

# Interprétation des contraintes

- $dHold > dAndUp2\_u + dLatchUp\_u$ 
  - ▶ Le temps de maintien de D doit être supérieur à la somme des temps maxima du front montant du « and » et de franchissement du latch
  
- $dAndUp2\_u + dLatchUp\_u < dNot1Down\_l + dNot2Up\_l + dXorDown1Up\_l + dAndDown1\_l$ 
  - ▶ La somme des temps maxima du front montant du « and » et de franchissement du latch doit être inférieure au temps minimal nécessaire entre le front descendant du « not 1 » et le front descendant du « and »

# Travaux futurs

- Utiliser des automates simulant le comportement complet des portes, et non juste la transition considérée dans notre système
  - ▶ Inutile dans notre cas
  - ▶ Susceptible de ralentir l'analyse
  - ▶ Mais nécessaire dans une perspective d'automatisation !